車載コックピット向けディスプレイ統合技術

Display Integration Technology for In-Vehicle Cockpit

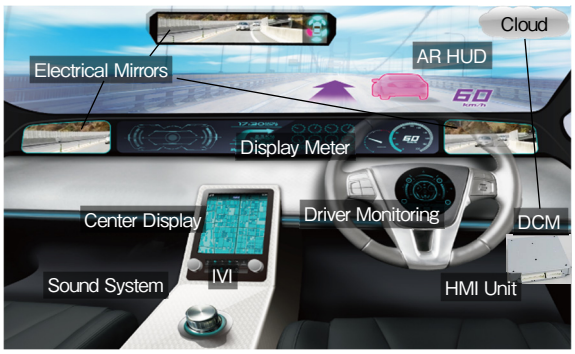
吉　田　直　史*　　坪　根　　隆*
Tadashi Yoshida　　Takashi Tsubone

現在，車載システムでは，ディスプレイを備えたHMI（Human Machine Interface）製品の数が増加のトレンドのなかにある．これら多数のディスプレイを活用し，より安心，安全，快適，便利な機能を実現していくため，コックピットに搭載される全てのディスプレイを統合的に扱える表示システムが求められている．仮想化技術を応用することで，複数のECU環境でも，ディスプレイを統合的に扱えるシステムを実現する．

Currently, the number of Human Machine Interface (HMI) products with displays used by in-vehicle systems is increasing. In order to utilize these many displays and realize safer, more comfortable and convenient functions, there is increased demand for a display system that can handle all the displays mounted in the cockpit in an integrated manner. We describe how we have applied virtualization technology to realize a system that can handle displays in an integrated manner even in various Electronic Control Unit (ECU) environments.

1. 複数ECU環境でのディスプレイ統合技術の必要性

近年，車室内にはメータクラスタ，カーナビゲーションシステム，ヘッドアップディスプレイ，電子ミラーなど，多数のディスプレイが搭載され，HMI（Human Machine Interface）アプリケーションが増加トレンドにある（**第1図**）．



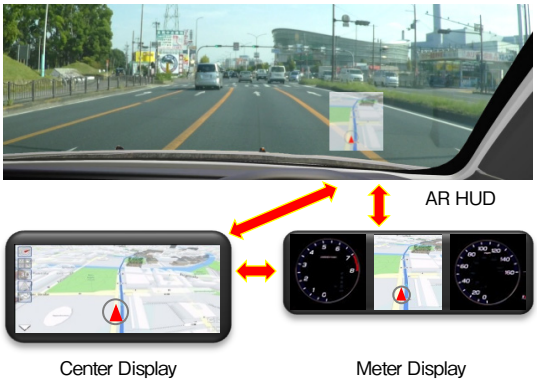
第1図　統合コックピットシステムのマルチディスプレイ
Fig. 1　Multiple displays on integrated cockpit system

これら多数のディスプレイを活用することで，より安心，安全，快適，便利な機能が実現できるようになる．

例えば，従来システムではADAS（Advanced driver-assistance systems）やV2X（注1）との連携によるドライバーへの注意喚起や警報などの危険通知は特定のディスプレイの固定位置への表示のみであったが，危険度やドライバー状態に応じて適切なディスプレイの適切な位置に，適切なタイミングで表示することで，危険通知の見落としをなくすことが可能となる（**第2図**）．

また，センターディスプレイに表示する地図や楽曲の情報をメータクラスタやヘッドアップディスプレイと連携して表示することにより，運転中に脇見をすることなく必要な情報を確認，操作することが可能となる（**第3図**）．

第2図　危険通知を最適な場所に表示
Fig. 2　Danger notifications in an optimized location



第3図　複数ディスプレイ間の表示コンテンツの連携
Fig. 3　Linked display contents across multiple displays

一方，複数のディスプレイ機器はそれぞれの特性に合わせて個別のECU（Electronic Control Unit）やOS（Operating

（注1）Vehicle to X．車とさまざまなモノ（車，歩行者，インフラ，ネットワークなど）をつなぐ通信技術．

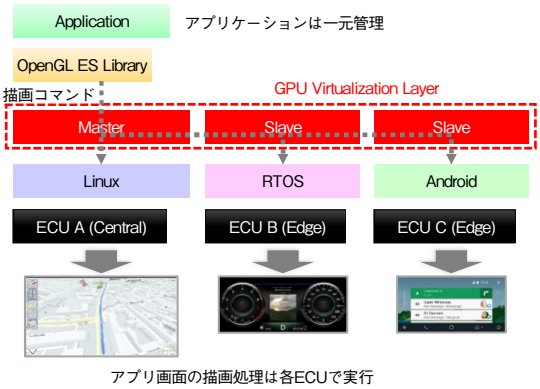
* オートモーティブ社　開発本部
R&D Div., Automotive Company

System）が採用されているため，複数ディスプレイの表示を統合的に扱うのは難しいという課題があった．例えば，複数のディスプレイ間でアプリ画面の共有を行うためには，個々のアプリがECU間で調停を行う必要があり，アプリ×ECUの数だけ開発が増加し，開発工数の爆発が発生する．また，開発が複雑化するため，不具合が混入しやすくなる．

そこで，上記の課題を解決するため，GPU（Graphics Processing Unit）仮想化技術を活用し，複数のECU/OSが混在する環境でも全てのディスプレイを1つのスクリーンとして統合的かつスケーラブルに扱えるようにすることで，複数ディスプレイを横断したHMIアプリの連携表示を容易に実現するディスプレイ統合技術を開発した．

2. 仮想化技術によるディスプレイ統合の実現

本技術では，汎用的なOpenGL ES (注2) ライブラリを用いて生成した描画コマンドをネットワーク経由で他のECUに転送することにより，他のECU上でGPU描画処理を実行することを可能にした．これにより，複数のECUを横断して表示されるHMIアプリの描画位置や内容を，1台のECU上で制御しつつ，GPUの能力に合わせて描画処理を各ECUに分散させることが可能となり，マルチECU環境で動作するHMIアプリの開発が容易になる（**第4図**）．



第4図　描画コマンド転送によるECUをまたいだGPU描画
Fig. 4　GPU drawing across ECUs by drawing command transfer

さらに，複数ディスプレイを1つの仮想スクリーン上に配置して統一的に扱うことで，アプリ開発者が複数ディスプレイにまたがるHMIの表示レイアウトを直感的かつ容易に制御することも可能となる．

また，描画コマンドの送受信を行うGPU仮想化レイヤーでECU/OS環境の違いを吸収することによって，HMIアプリはシステムを構成するECU/OSの違いを意識せずに利用することが可能となる．これにより，アプリケーションを変更することなく，システム構成をシングルECUとマルチECUでスケーラブルに変更することが可能となる．例えば，ECU数を減らしてコストダウンしたい場合にはシングルECU環境を選択し，全体制御はセントラルECUで行いつつ，負荷の高い描画処理はエッジ側で処理して負荷分散を図りたい場合にはマルチECU環境を選択すればよい．

GPU仮想化レイヤーの実装には，Linux (注3) カーネルのI/OインターフェースであるVirtIO (注4) およびVirtIO-GPUを活用する[1]．ただし，Hypervisorは必須ではない．また，GPU仮想化レイヤーはMaster-ECUで描画コマンドがどのECUで実行されるかを判断して転送を行い，Slave-ECUで転送コマンドを受信してGPU描画を実行する．

3. 本技術と従来技術の比較

本技術以外にも複数のディスプレイ間でアプリ画面を共有するための従来技術が存在する．従来技術には，描画された画面を動画エンコードしてリモートECUに送信する技術（Type A）や，3DCGのシーンを定義するシーングラフをリモートECUに送信して描画する技術（Type B）がある．以下に本技術と従来技術を比較した表を示す（**第1表**）．

Type Aは，送信側ECUでGPU描画した画面を動画エンコードしてリモートECUに送信する技術で，リモートECUのGPUが不要で負荷が小さく，転送データ量の変化が少ないという利点がある反面，送信側ECUの負荷が大きく，転送データ量も大きいという問題がある．

Type Bは，3DCGのシーンを定義するシーングラフをリモートECUに送信し，受信側でGPU処理可能な低レベルのAPIに変換して描画する技術である．描画処理をリモートGPUに負荷分散可能で，OpenGL ESより高レベルなAPIで記述するためコード量が少なく，シーンの変更点のみ転送する方式であるため転送データ量が非常に小さいという利点がある反面，専用APIを使用するため既存のOpenGL ESアプリやGUIツールキットをそのまま流用できず，アプリは専用の開発環境やライブラリを使って開発する必要があり，アプリの開発自由度が低いという問題がある．

一方で本技術は，OSのVirtIO対応が必要であるが，Linuxの場合は標準提供されており，汎用のOpenGL ESコマンドを送信してリモートECUでGPU描画を行う技術であるため，描画処理をリモートGPUに負荷分散可能で，Type Bほどで

（注2）OpenGL, OpenGL ESはSilicon Graphics, Inc.の米国およびその他の国における商標または登録商標．

（注3）Linus Torvalds氏の米国およびその他の国における登録商標または商標．

（注4）仮想化環境におけるデバイスのアクセス速度を向上させるI/Oフレームワーク．

第1表　本技術と従来技術の比較
Table 1　Comparison of this technology with the conventional technology

技術タイプ		汎用的な描画コマンドの送信（本技術）	動画エンコードされた画面の送信（TypeA）	シーングラフの送信（TypeB）
実現手段		OpenGL ESコマンドをリモートECUに送信して描画	GPUで描画された画面を動画エンコードしてリモートECUに送信	3DCGのシーンを定義するシーングラフをリモートECUに送信して描画
特徴	メリット	・リモートECUにGPU処理負荷を分散できる ・転送データ量が小さい ・既存のOpenGL ESアプリやGUIツールキットの流用が可能	・リモートECUにGPUが不要 ・転送データ量の変化が少ない	・リモートECUにGPU処理負荷を分散できる ・コード量が小さい ・転送データ量が非常に小さい
	デメリット	・OSにVirtIO対応が必要（Linuxでは標準提供）	・送信側ECUの処理負荷がが非常に大きい（GPU描画＋動画エンコード） ・転送データ量が大きい	・既存のOpenGL ESアプリやGUIツールキットをそのまま流用できない

はないが転送データ量が小さいという利点がある. さらに, 既存のOpenGL ESアプリやOpenGL ESでアクセラレートされたGUIツールキットの流用が可能であるという利点がある.

以上のように，本技術は従来技術と比較して優位性が大きいと言える.

4. 成果と展望

本技術により，複数ディスプレイを活用したHMIを効率的に実現することで，統合コックピットシステムのユーザーエクスペリエンスを大幅に向上することが期待できる.

今後は，本技術を活用したHMIの開発を支援する開発環境の整備を行っていくとともに，本技術を広く活用していただけるよう提案活動を進めていく.

参 考 文 献

[1] Matti Moell, “VirtIO-GPU,” GENIVI Alliance, https://at.projects.genivi.org/wiki/download/attachments/28412356/2018-10-11_GeniviBangalorTechSummit_Virtio_GPU.pdf, 参照 Apr. 20, 2021.